Time Series Analysis of Treasury Bill Rates

Yixuan (Gwen) Li, Mengze (Maisie) Tang, and Shengshi (Shirley) Yuan

Abstract

The US Treasury Bill rate is a critical indicator of the macro-economic condition. The interest rates of Treasury Bills inform the fundamentals of almost all other types of investment vehicles and thus reflect the overall market condition. When the economy slows down, the Fed lowers interest rates by purchasing Treasury securities to make borrowing more accessible and therefore stimulates economic growth; on the other hand, the interest rates are raised as a measure of controlling inflation. The objective of this study is to find an appropriate time series model to understand the variation in monthly rates for one-year Treasury Bills, and then forecast future rates based on the selected model. In this analysis, the data is collected on a monthly basis from January 1975 through October 2019 for Treasury Bills with one-year maturity, processed through transformation and differencing, and fitted with various time series models to find the best forecasting model. The analysis shows that ARIMA(4,1,5) is the best model to forecast future T-Bill rates and predict market condition for the first quarter of fiscal year 2020 to inform investment decisions for both individual and private equity fund investors.

Introduction

Treasury Bills (T-Bills) are government-issued short-term investment vehicles typically with a maturity of one-year or less. T-Bills are considered liquid, risk-free assets since investors are almost guaranteed to receive the face value upon maturity. Interest rates are critical because they not only are influenced by Fed's monetary policy to regulate the economy but also reflect investors' sentiment on events that affect the market. During periods of political and economic turmoil, for example, investors would withdraw from the riskier stock market and invest instead in the safer bond market, and vice versa. In this study, we analyze the monthly rates of T-Bills with one-year maturity spanning from January 1975 to October 2019, which cover three of the most notable economic booms and busts in the last 50 years, including the 1970s economic crash, the second worst recession since the Great Recession during 1981-1982, the "dot com bubble" in the 1990s, and the 2008-2009 economic recession as a result of the housing bubble. As the interest rate (indicated by monthly one-year T-Bill rates) closely tracks the macroeconomic events and market condition in the U.S., it can be used to analyze the economic cycle and forecast market condition in the near future in order to guide investment decisions for investors, and more specifically, private equity (PE) investors. Private equities engage in leveraged buyouts – a type of investment strategy where they borrow money from the bond market to invest in privately-held companies. Lower interest rates make funds more available to private equities, lower their cost of capital and thus increase their returns. As the rate hikes over the course of 2018 and 2019 raised concerns on increasing cost of capital for private equity investors, we seek to understand and forecast the T-Bill rate, an important indicator for interest rate, for the first quarter of 2020. To that end, we analyze the fluctuation in the monthly 1-year T-Bill rates by fitting the processed data with various time series models including ETS(A,A_d,N), ARIMA(4,1,5), ARMA(4,5)+Garch(1,1), Neural Network, SETAR(1,4), LSTAR and Regression to find the best-fitting model to forecast the future T-Bill rates. Our result shows that ARIMA(4,1,5) provides the best forecast, which will be used to predict future rates and establish guideline for fund investors.

I. Data

The monthly 1-year Treasury Bill rates data is sourced from the Federal Reserve Bank of St. Louis. (Fred). As shown in *figure 1*, following the economic crash in the early 1970s, a period marked by low interest rates in order to invigorate the economy, the Fed raised rates to combat inflation throughout the remainder of the 1970s. This tightening, along with the economic and geopolitical changes, spiked the one-year T-Bill rate to nearly 15%. The extremely high interest rates then fell sharply thanks to

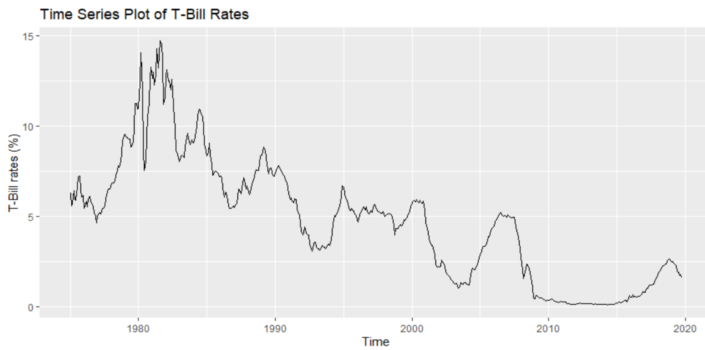
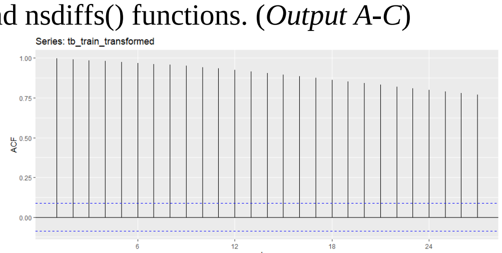


Figure 1. Time Series Plot of T-Bill Rates (Jan 1975 – Oct 2019)

the Fed's action in 1980 when the U.S. entered into another economic recession and unemployment rate peaked. This was followed by a period of active adjustments by Fed which led to the highest interest rate in the last half century to control high inflation. The interest rate remained high until the 1981-1982 recession when the Fed had to cut interest rates to allow recovery, and the resulting low rates lasted through the 2000s. The interest rate reached a historical low following the 2008-2009 recession and remained around 2% until the several interest rate hikes by Fed just a year ago (Business Insider). Based on *figure 1*, we find that, consistent with what we expect to see with business cycles, there is no trend, no seasonality but significant cyclicality in the data collected. Our data contains 538 entries in total, where the first 516 entries are designated training set and the last 22 are used to test models.

II. Analysis and Models Applied

Data Transformation and Obtaining Stationarity. Based on *figure 1*, the data demonstrates unstable variance and thus we apply transformation using optimal lambda (0.235) to obtain stable variance. The ACF and PACF of the transformed training data shown in *Figure 2 (a) and (b)* respectively suggests non-stationarity, and thus we apply first order differencing to the transformed training data. The result of KPSS test after first-order differencing indicates the data is then stationary, which is consistent with result from ndiffs() and nsdiffs() functions. (*Output A-C*)



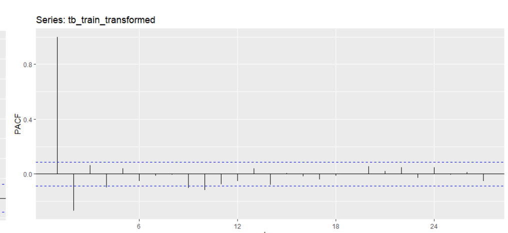
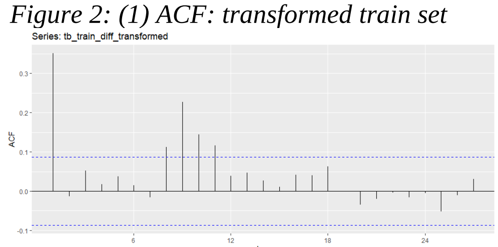
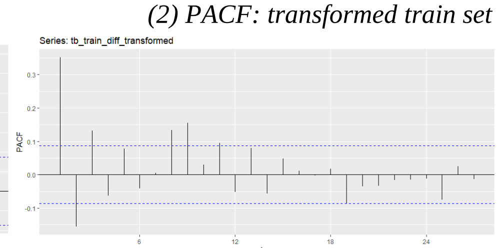


Figure 2: (1) ACF: transformed train set *(2) PACF: transformed train set*





(3) ACF: transformed differenced train set *(4) PACF: transformed differenced train set*

1. Additive Damped Trend Method with Additive Errors. Since the last recession, the T-Bill rates have remained low, therefore we apply ETS (A, Ad, N) model as it allows us to weigh the more recent data more highly. The obtained model is (see *Output D*):

$$y_t = l_{t-1} + \phi b_{t-1} + \varepsilon_t$$
$$l_t = l_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$$
$$b_t = \phi b_{t-1} + \beta^* \varepsilon_t$$

Based on Ljung-Box test and the ACF plot in Output D, the model does not produce white noise residuals and therefore is insufficient. Thus, we will not further consider this model an appropriate fit for forecasting future T-Bill rates.

2. ARIMA(4, 1, 5). The ACF plot of original data suggests autocorrelation, implying that future data can be predicted by lagged values; and because the data is non-stationary, we anticipate that ARIMA could be sufficient. Based on the model given by auto.arima function, we looped through all possible neighbor models to select the one with lowest AIC (see *Output E*):

$$(1-\phi_1B-\phi_2B^2-\phi_3B^3-\phi_4B^4)(1-B)y_t = c+(1+\Theta_1B+\Theta_2B^2+\Theta_3B^3+\Theta_4B^4+\Theta_5B^5)\varepsilon_t$$

According to this model, the T-Bill rate increases slightly throughout 2018 and then stays relatively flat in 2019, closely resembling the true T-Bill rate near the end of 2019 (see *figure 3*). As shown in *Output E*, the ARIMA(4, 1, 5) produces white noise residuals and AIC of -919.8.

3. ARMA(4, 5)+GARCH(1, 1). Since the GARCH model can be used to capture the volatility in financial data, we apply ARMA+GARCH to the data. We first examine the ARCH effect using ACF of residuals and PACF of residuals squared (see *Output F*). The ACF of residuals shows no spike within first 6 lags, and the PACF of residuals squared shows some significant spikes, indicating the existence of ARCH effect. We then run a "for" loop to fit the ARMA(4,5)+GARCH(i,j) with i=1,2,3,4 and j=0,1,2,3. The smallest AIC (-2.077399) comes from ARMA(4,5)+GARCH(1,1). The resulting model is in *Output F* and the equation for our model is:

$$r_t = 0.0016a_t + 0.1344r_{t-1} - 0.3039r_{t-2} + 0.4485r_{t-3} + 0.3968r_{t-4} + 0.2482a_{t-1} + 0.2636a_{t-2} - 0.1897a_{t-3} - 0.0115a_{t-4}$$

$$\sigma_t^2 = 0.0013 + 0.3926a_{t-1}^2 + 0.4985\sigma_{t-1}^2$$

$$y_t = \sigma_t\varepsilon_t$$

where $\{\varepsilon t\}$ is an i.i.d. normal time series with mean zero and variance one. The residuals plots and Ljung-Box test (*Output F*) indicate that the residuals can be considered as white noise. As shown in *Figure 3*, the ARMA+GARCH model gives reasonable forecast results, with forecasted T-Bill rates fluctuating downward in the second half of 2019.

4. Neural Network. Since the data does not have seasonality, we simply try a Neural Network model using nnetar(), which gives a 1-1-1 neural network with lag of 14. The neural network model relies on randomness, with a different seed we may obtain different results both for the AIC and for the prediction. The parameters of the model are shown in *Output G*. As shown in *Figure 3*, the forecast of the Neural Network does not perform very well. Moreover, the p-value of the Ljung-Box test of the residuals is 0.9968, indicating that the residuals cannot be considered as white noise. Thus, we will not further consider this model in forecasting T-Bill rates.

5. SETAR(1, 4). SETAR model allows us to account for any potential nonlinear relationships in the data. Since T-Bill rates data does not have seasonality, we try various SETAR models with lag 1 and different combinations of threshold variables. The SETAR(1, 4) gives the lowest AIC. The parameters of the fitted model are shown in *Output H*. The equation for our model is:

$$x_t = 0.0340 + 1.1198x_{t-1} - 0.3970x_{t-2} + 0.4970x_{t-3} - 0.1988x_{t-4} + a_t \;\; if \; x_{t-2} < 0.2215$$

$$x_t = 0.0103 + 1.5660x_{t-1} - 0.7509x_{t-2} + 0.2613x_{t-3} - 0.0744x_{t-4} + a_t \;\; if \; x_{t-2} \geq 0.2215$$

As shown in *figure 3*, the forecast of the fitted SETAR model gives a straight downward line and fails to capture the fluctuations in the data. The result of Ljung-Box test indicates that residuals can be considered as white noise (*Output H)*, but the model does not provide good forecast results.

6. LSTAR. LSTAR model is tried to allow for a higher degree of flexibility. The parameters of the fitted model are shown in *Output I*. As shown in *figure 3*, the forecast of the fitted LSTAR model is close to SETAR and shows a downward trend but fails to capture the fluctuation in the data. Furthermore, the p-value of the Ljung-Box test of the residuals is 0.0110, indicating that the residuals cannot be considered as white noise and thus making the LSTAR model insufficient.

7. Time Series Regression. Time series regression model is applied since the interest rate is influenced by a number of macroeconomic indicators such as unemployment rate and stock market return, as measured by monthly return of Dow Jones Industrials Index. Since equity and bond are alternative investment classes, we expect to see more investment flow into bond market when the stock market is not doing well. Additionally, we expect to see lower interest rate to stimulate the economy when unemployment rate is high. The fitted model is:

$$T - Bill\ rates = 6.002 - 0.428\ Unemployment\ Rate - 2.512 \times 10^{-4}\ Stock\ Return$$

The adjusted R^2 is 0.793, which indicates significant predictive power. However, the residuals cannot be considered as white noise (*Output J*). Therefore, this model is insufficient.

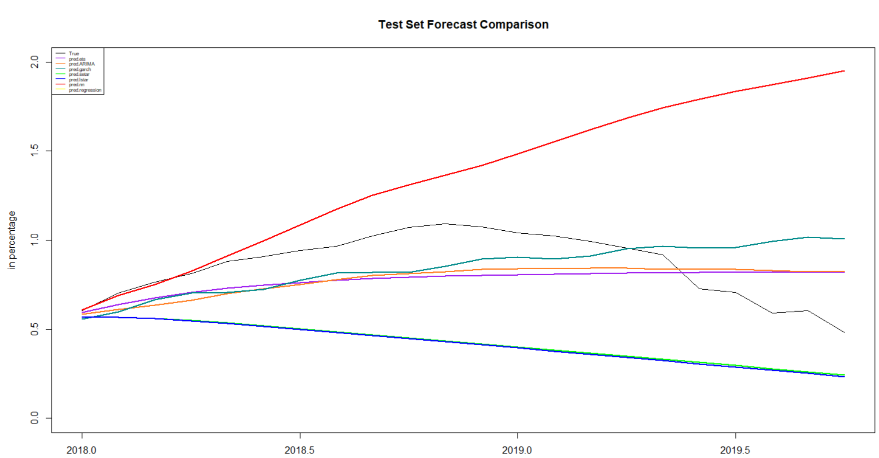


Figure 3. Test Set Forecast Comparison

III. Discussion

	ETS	ARIMA	GARCH	SETAR	LSTAR	NN	REG
AIC	914.2691	-919.7686	-2.077399	-2402.722	-2400.388	N/A	1087.65
MAE	0.176938	0.176316	0.052220	0.068262	0.441605	0.499548	2.90366
MAPE	21.42893	21.70815	108.2037	8.312612	49.96487	69.30398	354.622
RMSE	0.194894	0.190120	0.072367	0.095785	0.478271	0.686494	2.91154

Table 1

Based on above table, although some of the metrics don't agree, it is clear that overall, ARIMA and GARCH, both with white noise residuals, provide the most outstanding forecast accuracy among the 7 models applied. ARIMA has much lower AIC and MAPE while GARCH has lower MAE and RMSE. However, based on forecast plot in *figure 3*, ARIMA provides much closer predictions to true data towards the end of the forecast period and well captures the slightly downward trend starting in 2019, which GARCH fails to capture. Thus, we select ARIMA(4,1,5) as the best model to forecast T-Bill rates in the first quarter of 2020.

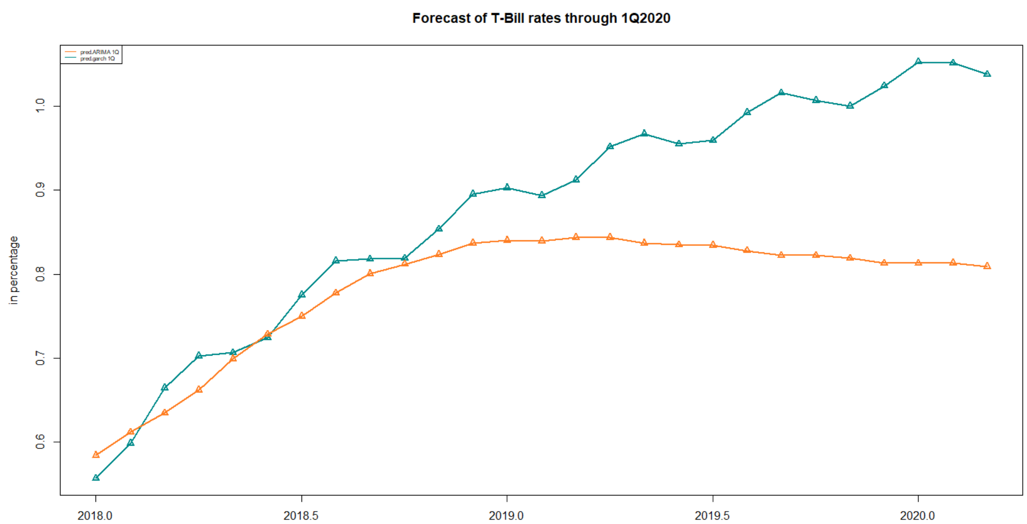


Figure 4. T-Bill Rates Forecast though 1Q 2020

Based on the forecast using ARIMA(4,1,5) model, the one-year T-Bill rates in 1Q2020 stay in the low range of 0.5%-1.0%, even declining slightly compared to the 2018 and 2019 level. Lower interest rates signal easier access to funds and stimulates economic activities. This bodes well for the overall economy and thus the investment activities of private equities, who are able to borrow at low rates to invest in the equity market to generate higher returns. Therefore, despite the several interest rate hikes by Fed over the past two years, we expect future T-Bill rates to remain at low levels compared to historical figures and even decline slightly, and thus we recommend further fund raising and transaction activities for private funds in the first quarter of 2020. For individual investors, we recommend shorting the U.S. bond market and buying in the equity market.

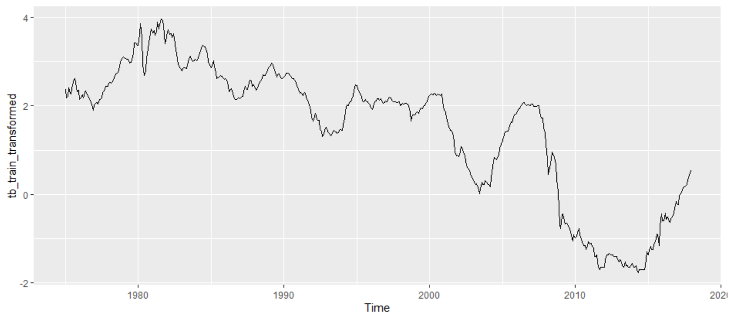
Reference

“Here's What the Major Interest Rate Cycles since the 1970s Have Looked Like.” *Business Insider*, Business Insider, 19 Dec. 2015, https://www.businessinsider.com/every-interest-rate-cycle-since-1970s-2015-12#-14.

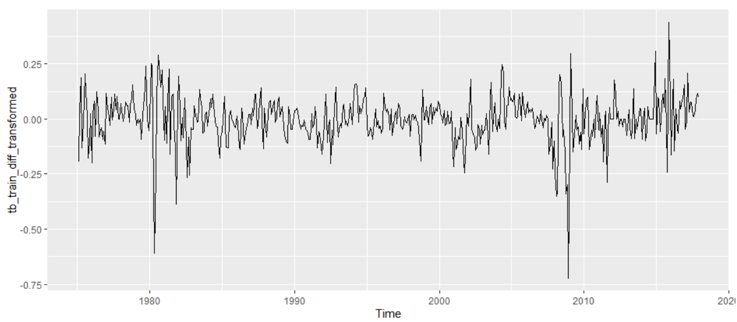
“1-Year Treasury Bill: Secondary Market Rate.” *FRED*, 2 Dec. 2019, https://fred.stlouisfed.org/series/TB1YR.

Appendix

Output A. Plot of Transformed Data



Output B. Plot of Transformed and Differenced Data



Output C. KPSS test after first-order differencing

```
#######################
# KPSS Unit Root Test #
#######################

Test is of type: mu with 6 lags.

Value of test-statistic is: 0.1189

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739


nsdiffs(tb_train_transformed)

#0

ndiffs(tb_train_transformed)
#1, confirming results of kpss test below

ndiffs(tb_train_diff_transformed)
#0
```

Output D. Output Results from ETS (A, Ad, N)

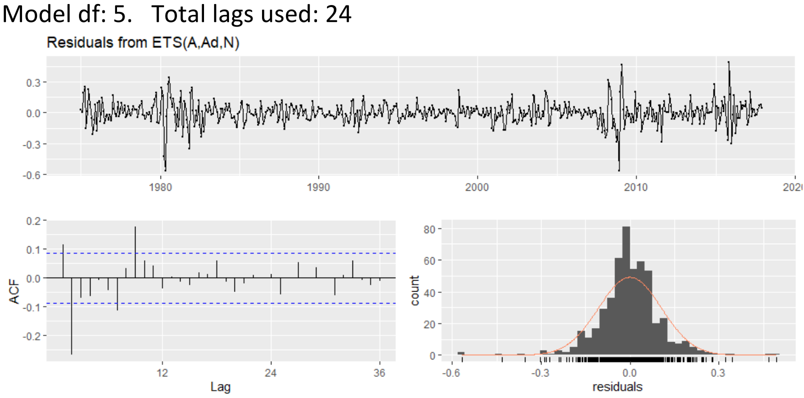
```
ETS(A,Ad,N)

Call:
 ets(y = tb_train_transformed)

  Smoothing parameters:
    alpha = 0.9999
    beta  = 0.3792
    phi   = 0.8

  Initial states:
    l = 2.6086
    b = -0.3237

  sigma:  0.1064

     AIC     AICc      BIC
918.0718 918.2368 943.5484

Training set error measures:
                        ME      RMSE        MAE       MPE     MAPE
Training set -0.0003783696 0.1059194 0.07337714 -6.090054 12.46329
                  MASE      ACF1
Training set 0.1743699 0.1148923
```

accuracy(fit4,tb_test_transformed)

```
                        ME      RMSE        MAE       MPE
Training set -0.0003783696 0.1059194 0.07337714 -6.090054
Test set      0.0872774261 0.1948937 0.17693830  5.662131
                 MAPE      MASE      ACF1 Theil's U
Training set 12.46329 0.1743699 0.1148923        NA
Test set     21.42893 0.4204676 0.7956015  2.637266
```

Ljung-Box test

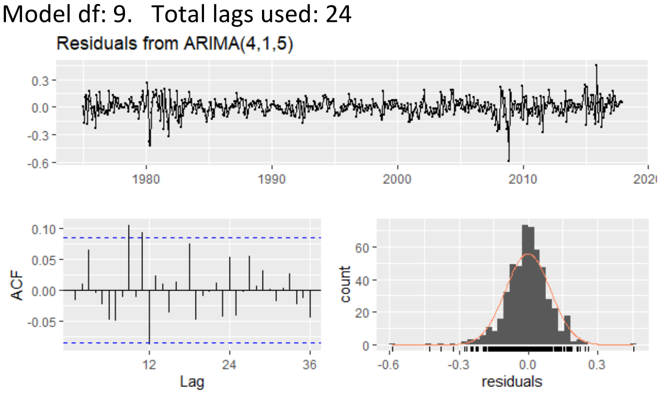
```
data:  Residuals from ETS(A,Ad,N)
Q* = 82.005, df = 19, p-value = 8.372e-10

Model df: 5.   Total lags used: 24
```



Output E. Output Results from ARIMA(4,1,5)

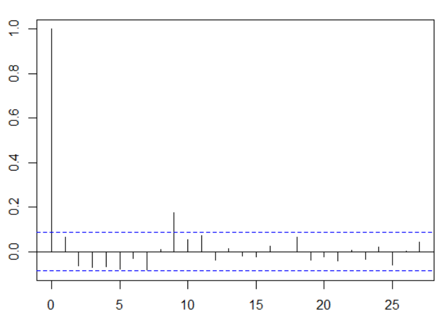
summary(fit5)

```
Series: tb_train_transformed
ARIMA(3,1,4)(1,0,1)[12]

Coefficients:
         ar1     ar2      ar3      ma1      ma2    ma3     ma4     sar1
      1.0159  0.5306  -0.6261  -0.5940  -1.0680  0.500  0.3196  -0.5240
s.e.  0.2088  0.3253   0.1489   0.2153   0.2429  0.082  0.1151   0.3289
        sma1
      0.4599
s.e.  0.3406

sigma^2 estimated as 0.009723:  log likelihood=466.51
AIC=-913.02   AICc=-912.58   BIC=-870.58

Training set error measures:
                       ME       RMSE       MAE       MPE     MAPE     MASE
Training set -0.001616025 0.09764333 0.0699446 -5.812426 12.15757 0.166213
                    ACF1
Training set 0.003920429
accuracy(r.hat6,tb_test_transformed)
                 ME     RMSE       MAE      MPE     MAPE      ACF1
Test set 0.08242699 0.1900967 0.1762899 5.313498 21.70653 0.7946345
         Theil's U
Test set   2.64928

Ljung-Box test

data:  Residuals from ARIMA(4,1,5)
Q* = 28.355, df = 15, p-value = 0.01945

Model df: 9.   Total lags used: 24
```

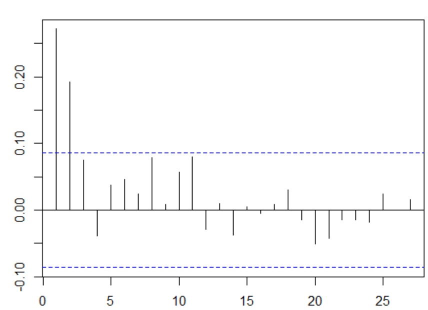


Output F. Output Results from ARMA(4, 5) + Garch(1, 1)

Testing for ARCH effect:



(1)ACF of residuals



(2) PACF of residuals squared

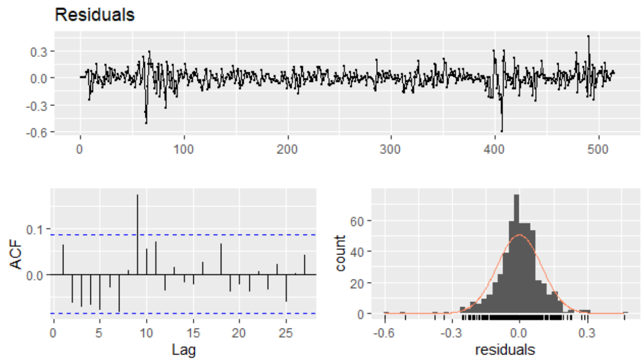
summary(fit777)

```
Title:
 GARCH Modelling

Call:
 garchFit(formula = ~arma(4, 5) + garch(1, 1), data = tb_train_diff_transformed,
    trace = F)

Mean and Variance Equation:
 data ~ arma(4, 5) + garch(1, 1)
<environment: 0x00000000057fa558>
 [data = tb_train_diff_transformed]

Conditional Distribution:
 norm

Coefficient(s):
        mu        ar1        ar2        ar3        ar4        ma1
 0.0015773  0.1343666 -0.3038570  0.4484546  0.3967904  0.2482211
       ma2        ma3        ma4        ma5      omega     alpha1
 0.2636217 -0.1896966 -0.5237061  0.0115338  0.0012730  0.3926320
     beta1
 0.4984570

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.0015773   0.0026153    0.603  0.54644
ar1     0.1343666   0.1905589    0.705  0.48074
ar2    -0.3038570   0.1464789   -2.074  0.03804 *
ar3     0.4484546   0.1080556    4.150 3.32e-05 ***
ar4     0.3967904   0.1488871    2.665  0.00770 **
ma1     0.2482211   0.1937059    1.281  0.20004
ma2     0.2636217   0.1067836    2.469  0.01356 *
ma3    -0.1896966   0.1021319   -1.857  0.06326 .
ma4    -0.5237061   0.1202034   -4.357 1.32e-05 ***
ma5     0.0115338   0.0789760    0.146  0.88389
omega   0.0012730   0.0004254    2.992  0.00277 **
alpha1  0.3926320   0.0812186    4.834 1.34e-06 ***
beta1   0.4984570   0.0947069    5.263 1.42e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 547.9302    normalized:  1.063942

Description:
 Tue Dec 03 22:20:17 2019 by user: Gwen


Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  19.00472  7.467542e-05
 Shapiro-Wilk Test  R    W      0.9897029 0.001126834
 Ljung-Box Test     R    Q(10)  13.00032  0.2236538
 Ljung-Box Test     R    Q(15)  17.31095  0.3006183
 Ljung-Box Test     R    Q(20)  22.99984  0.2888022
 Ljung-Box Test     R^2  Q(10)  7.165716  0.7097132
 Ljung-Box Test     R^2  Q(15)  17.21464  0.3061959
 Ljung-Box Test     R^2  Q(20)  20.79116  0.4095154
 LM Arch Test       R    TR^2   14.19447  0.2884626
```

```
Information Criterion Statistics:
      AIC       BIC       SIC      HQIC 
-2.077399 -1.970265 -2.078632 -2.035413
```

Box.test(fit777@fitted)

```
# Box-Pierce test

# data:  fit777@fitted

# X-squared = 16.364, df = 1, p-value = 5.227e-05
```



Output G. Output Results from Neural Network

summary(fit.nn)

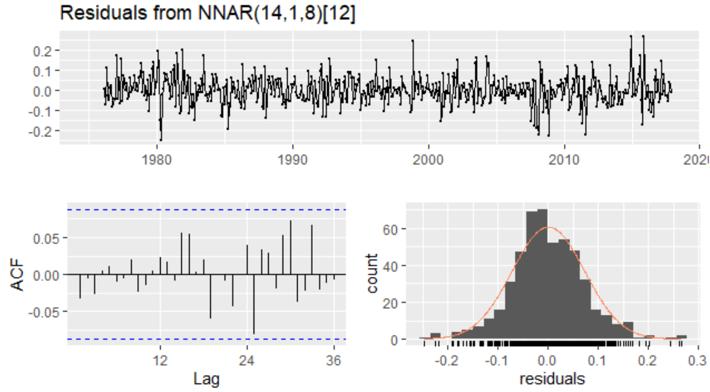
```
          Length Class        Mode     
x         516    ts           numeric  
m           1    -none-       numeric  
p           1    -none-       numeric  
P           1    -none-       numeric  
scalex      2    -none-       list     
size        1    -none-       numeric  
subset    516    -none-       numeric  
model      20    nnetarmodels list     
nnetargs    0    -none-       list     
fitted    516    ts           numeric  
residuals 516    ts           numeric  
lags       14    -none-       numeric  
series      1    -none-       character
method      1    -none-       character
call        2    -none-       call
```

accuracy(pred.nn,tb_test_transformed)

```
                        ME       RMSE        MAE        MPE      MAPE
Training set  0.0001974124 0.07235806 0.05513582  -3.910335  8.847339
Test set     -0.5308317877 0.72675199 0.53557805 -73.245193 73.926558
                  MASE        ACF1 Theil's U
Training set 0.1310221 -0.03272796        NA
Test set     1.2727216  0.86170616  11.39636
```

Box.test(fit.nn$residuals,lag = 9, type='Ljung')

```
	Box-Ljung test

data:  fit.nn$residuals
X-squared = 1.55, df = 9, p-value = 0.9968
```



Output H. Output Results from SETAR (1,4)

Non linear autoregressive model

SETAR model (2 regimes)

Coefficients:

Low regime:

```
   const.L      phiL.1      phiL.2      phiL.3
 0.03404341  1.11976645 -0.39701076  0.49697380
     phiL.4
-0.19884213
```

High regime:

```
   const.H      phiH.1      phiH.2      phiH.3
-0.01030320  1.56599378 -0.75091379  0.26129486
     phiH.4
-0.07437634
```

Threshold:

-Variable: Z(t) = + (0) X(t)+ (1)X(t-1)+ (0)X(t-2)+ (0)X(t-3)

-Value: 0.2215

Proportion of points in low regime: 22.27%   High regime: 77.73%

Residuals:

```
       Min        1Q     Median        3Q
-0.5962663 -0.0429860 -0.0010787  0.0560184
       Max
 0.4081837
```

Fit:

residuals variance = 0.009104, AIC = -2403, MAPE = 10.06%

Coefficient(s):

```
         Estimate  Std. Error  t value  Pr(>|t|)
const.L  0.034043    0.017356   1.9614 0.0503761
```

phiL.1   1.119766   0.072007 15.5509 < 2.2e-16

phiL.2 -0.397011   0.110797 -3.5832 0.0003722

phiL.3   0.496974   0.109927   4.5210 7.675e-06

phiL.4 -0.198842   0.069525 -2.8600 0.0044113

const.H -0.010303   0.014560 -0.7076 0.4795043

phiH.1   1.565994   0.053693 29.1657 < 2.2e-16

phiH.2 -0.750914   0.097655 -7.6894 7.759e-14

phiH.3   0.261295   0.097488   2.6803 0.0075961

phiH.4 -0.074376   0.054419 -1.3667 0.1723146

const.L .

phiL.1 ***

phiL.2 ***

phiL.3 ***

phiL.4 **

const.H

phiH.1 ***

phiH.2 ***

phiH.3 **

phiH.4

Signif. codes:

0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Threshold

Variable: Z(t) = + (0) X(t) + (1) X(t-1)+ (0) X(t-2)+ (0) X(t-3)

Value: 0.2215

[1] -2402.722

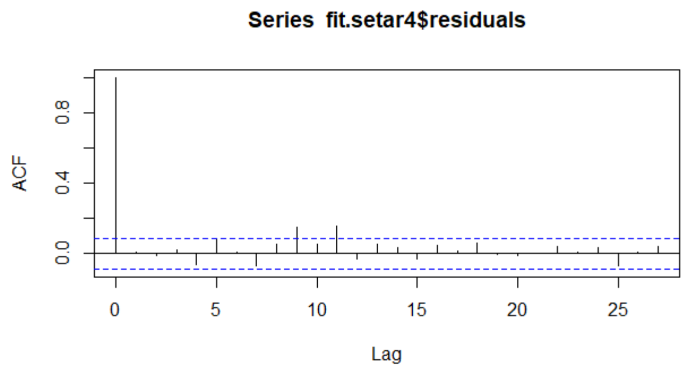
Box-Ljung test

data: fit.setar4$residuals

X-squared = 22.769, df = 10, p-value =

0.01163

[1] 0.2237386



Output I. Output Results from LSTAR
summary(fit.lstar)

```
Non linear autoregressive model

LSTAR model
Coefficients:
Low regime:
   const.L      phiL.1      phiL.2      phiL.3      phiL.4
 0.03685584  1.12405152 -0.40364200  0.50514059 -0.20244340

High regime:
   const.H      phiH.1      phiH.2      phiH.3      phiH.4
-0.04747295  0.43964946 -0.34277330 -0.24758529  0.12972216

Smoothing parameter: gamma = 100

Threshold
Variable: Z(t) = + (0) X(t) + (1) X(t-1)+ (0) X(t-2)+ (0) X(t-3)

Value: 0.2282

Residuals:
       Min          1Q      Median          3Q         Max
-0.59622255 -0.04288145 -0.00077039  0.05578939  0.40767390

Fit:
residuals variance = 0.00911,  AIC = -2400, MAPE = 10.27%

Coefficient(s):
         Estimate  Std. Error  t value  Pr(>|z|)
const.L   0.036856    0.018468   1.9957 0.0459704 *
phiL.1    1.124052    0.071508  15.7193 < 2.2e-16 ***
phiL.2   -0.403642    0.110477  -3.6536 0.0002586 ***
phiL.3    0.505141    0.110268   4.5810 4.627e-06 ***
phiL.4   -0.202443    0.069419  -2.9163 0.0035424 **
const.H  -0.047473    0.023864  -1.9893 0.0466652 *
phiH.1    0.439649    0.089386   4.9185 8.720e-07 ***
phiH.2   -0.342773    0.147386  -2.3257 0.0200354 *
phiH.3   -0.247585    0.147187  -1.6821 0.0925458 .
phiH.4    0.129722    0.088085   1.4727 0.1408328
gamma   100.000004  257.902498   0.3877 0.6982059
th        0.228239    0.024692   9.2433 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Non-linearity test of full-order LSTAR model against full-order AR model
 F = 3.2707 ; p-value = 0.011567
```

Threshold
Variable: Z(t) = + (0) X(t) + (1) X(t-1)+ (0) X(t-2)+ (0) X(t-3)

accuracy(pred.lstar, tb_test_transformed)

```
            ME      RMSE       MAE      MPE     MAPE      ACF1
Test set 0.4416052 0.4782705 0.4416052 49.96487 49.96487 0.7982632
       Theil's U
Test set  5.767757
```

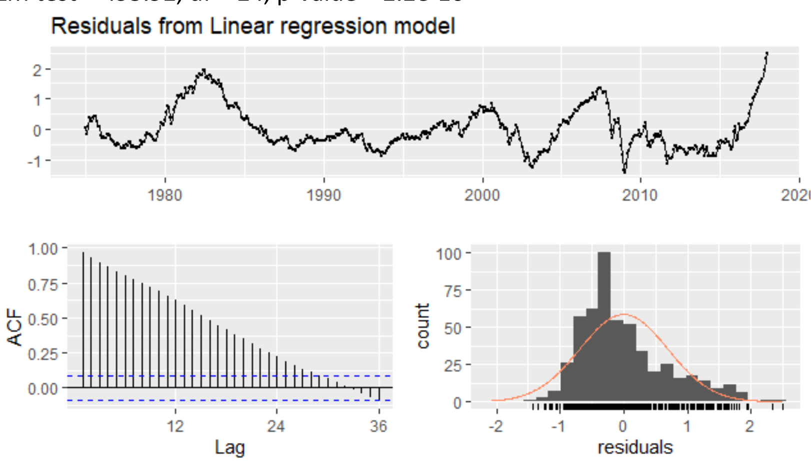
Box-Ljung test

data: fit.lstar$residuals
X-squared = 21.408, df = 9, p-value = 0.01096

Output J. Output Results from Time Series Regression

summary(reg1)

```
Call:
tslm(formula = tb_train_transformed ~ unemployment_train1 + return_train1)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.4246 -0.4580 -0.1920  0.3473  2.5005 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          6.002e+00  1.547e-01   38.80   <2e-16 ***
unemployment_train1 -4.282e-01  2.063e-02  -20.75   <2e-16 ***
return_train1       -2.512e-04  5.688e-06  -44.17   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6908 on 513 degrees of freedom
Multiple R-squared:  0.7937,	Adjusted R-squared:  0.7929 
F-statistic: 986.6 on 2 and 513 DF,  p-value: < 2.2e-16

ME    RMSE      MAE      MPE     MAPE      ACF1 Theil's U
Test set 2.903657 2.91154 2.903657 354.6218 354.6218 0.4515302  38.28549

Breusch-Godfrey test for serial correlation of order up to 24

data:  Residuals from Linear regression model
LM test = 495.92, df = 24, p-value < 2.2e-16
```



R Code Used to Produce Outputs

```
#install.packages("fBasics")
#install.packages("fGarch")
#install.packages("fpp2")
#install.packages("tsensembler")

library(fBasics)
library(fGarch)
library(fpp2)
#library(tsensembler)


library(data.table)
tb.rates=read.csv("tb_rates.csv",header=T)
tb <- ts(tb.rates$tb_rates, frequency=12, start=c(1975, 1))
tb_train1 <- window(tb, end = c(2017,12))
tb_test1<-window(tb, start=c(2018, 1))


#Original time series plot of data
autoplot(tb,ylab = "T-Bill rates (%)", main = "Time Series Plot of T-Bill Rates")

#Transforming data to obtain stable variance
lambda1<-BoxCox.lambda(tb_train1)
tb_train_transformed <- BoxCox(tb_train1, lambda =lambda1)
tb_test_transformed <- BoxCox(tb_test1, lambda =lambda1)
Tb_transformed <- BoxCox(tb, lambda =lambda1)

#Plotting Transformed data
autoplot(tb_train_transformed)
ggAcf(tb_train_transformed)
ggPacf(tb_train_transformed)

#Differencing to obtain stationarity
tb_train_diff_transformed=diff(tb_train_transformed)
tb_test_diff_transformed=diff(tb_test_transformed)
#Plotting Transformed and differenced data
autoplot(tb_train_diff_transformed)
ggAcf(tb_train_diff_transformed)
ggPacf(tb_train_diff_transformed)

#Testing order of differencing:
nsdiffs(tb_train_transformed)
#0
ndiffs(tb_train_transformed)
#1, confirming results of kpss test below

#KPSS test for stationarity
#install.packages("urca")
library(urca)
summary(ur.kpss(tb_train_diff_transformed))
ndiffs(tb_train_diff_transformed)
#0
```

Basic Models

```
mean_tb<-meanf(tb_train_diff_transformed, h=22)
naive_tb<-naive(tb_train_diff_transformed, h=22)
snaive_tb<-snaive(tb_train_diff_transformed, h=22)
drift_tb<- rwf(tb_train_diff_transformed, drift=TRUE, h=22)
diff_tb_transformed <- diff(BoxCox(tb, lambda =lambda1))
length(diff_tb_transformed)
length(tb_train_diff_transformed)
tb.test<- window(diff_tb_transformed,  start=c(2018, 1))
plot(diff_tb_transformed, ylab = "Forecast of Differenced Rates", main = "Forecast Re
sults from Simple Forecast Methods")
lines(mean_tb$mean, col = "red")
lines(naive_tb$mean, col = "blue")
lines(snaive_tb$mean, col = "yellow")
lines(drift_tb$mean, col = "green")
lines(tb.test,col = "orange")
legend("topright", legend = c("mean","naive","snaive","drift","true data"),
       col = c("red","blue","yellow","green","orange"),lty = 1:2,cex = 0.3)
```

ETS model

```
object<-ets(tb_train_transformed)
summary(object)
library(forecast)
fit4<-forecast.ets(object,
               h=22,
               level=c(80,95), fan=FALSE,
               simulate=FALSE, bootstrap=FALSE,
               npaths=5000, PI=TRUE,
               lambda=object$lambda, biasadj=FALSE)
autoplot(fit4, ylab = "Transformed T-Bill Rates")
# plot(tb_train_transformed,lwd=3,col="black")
# lines(object$state,lwd=1,col="red")
accuracy(fit4,tb_test_transformed)
summary(fit4)
#AIC     AICc      BIC
#914.2691 914.3140 927.1326
ets.aicc <- 914.3140
ets.aic <- 914.2691
ets.rmse <- 0.1948937
ets.mae <- 0.17693830
ets.mape <- 21.42893
ets.mase <- 0.4204676
checkresiduals(fit4)
#Ljung-Box test

#data:  Residuals from ETS(A,N,N)
#Q* = 135.5, df = 22, p-value < 2.2e-16

#Model df: 2.   Total lags used: 24
```

ARIMA

```
#AUTO ARIMA
fit5 <- auto.arima(tb_train_transformed)
summary(fit5)
fit5 %>% forecast(h=22) %>% autoplot(include=538)
```

```r
AIC(fit5)
# ARIMA(3,1,4)(1,0,1)[12]
#[1] -913.0205

ggAcf(tb_train_transformed)
ggPacf(tb_test_transformed)
ggAcf(tb_train_diff_transformed)

# ARIMA LOOP ##
# aic_all = c()
#
# for (i in 0:6){
#    for (j in 0:6){
#      for (m in 0:3){
#        for (n in 0:3){
#          fit=arima(tb_train_transformed,order = c(i,1,j),seasonal=c(m,0,n))
#          aic = fit$aic
#          aic_tmp = c(i,j,m,n,aic)
#          aic_all = rbind(aic_all,aic_tmp)
#        }
#      }
#    }
#  }
#
#  z=which.min(aic_all[,5])
#  combos=cbind(1:784,aic_all)
#  combos[z,]

#this chooses an ARIMA(4,1,5) model
#AIC = -919.7687

fit6=arima(tb_train_transformed,order=c(4,1,5))
fit6 %>% forecast(h=22) %>% autoplot(include=538)
r.hat6<-predict(fit6,22)$pred
accuracy(r.hat6,tb_test_transformed)
AIC(fit6)
#[1] -919.7687
arima.aic <- AIC(fit6)
arima.rmse <- 0.1901198
arima.mae <- 0.1763157
arima.mape <- 21.70815



checkresiduals(fit6)

# Ljung-Box test
#
# data:  Residuals from ARIMA(4,1,5)
# Q* = 28.349, df = 15, p-value = 0.01948
#
# Model df: 9.   Total lags used: 24
```

GARCH model

```
###### GARCH model ######

#fit the ARCH model
#install.packages("fGarch")
library(fGarch)
fit.arch = garchFit(~garch(1,0),data=tb_train_transformed,trace=F)
arch.mae<-mean(abs(fit.arch@residuals))
arch.mse<-mean((fit.arch@residuals)^2)
arch.mape<-mean(abs((fit.arch@residuals)/tb_train_transformed))*100
arch.rmse <- sqrt(mean((fit.arch@residuals)^2))

cbind(arch.mae,arch.rmse,arch.mape)
#    arch.mae arch.rmse arch.mape
#[1,] 1.148736  1.628596  240.1895
summary(fit.arch)
arch.aic <- 1.884714
#    AIC      BIC      SIC      HQIC
#1.884714 1.909401 1.884647 1.894388

#fit the GARCH model
fit.garch = garchFit(~garch(1,1),data=tb_train_transformed,trace=F)
summary(fit.garch)
#AIC      BIC      SIC      HQIC
#1.888563 1.921478 1.888444 1.901461 GARCH is unnecessary, the p-value for beta is 0.
457
mae.garch<-mean(abs(fit.garch@residuals))
mse.garch<-mean((fit.garch@residuals)^2)
#mape.garch<-mean(abs((fit.garch@residuals)/tb_test_transformed))*100
#cbind(mae.garch,mse.garch,mape.garch)
# mae.garch mse.garch mape.garch
#[1,]  1.148765   2.65285   240.2336


# r.hat.garch<-predict(fit.garch,134)$pred
# accuracy(r.hat.garch,tb_test_transformed)

#Obtain standardized residuals
b=residuals(fit.arch,standardize=T)
plot(b,type='l')

#the residual plot show that itâ€™s not a good fit
#To check adequacy, check both Ljung-Box of the residuals and the residuals squared
Box.test(b,10,type='Ljung')
#p-value is extremely small
#To check the adequacy of the mean equation
Box.test(b^2,10,type='Ljung')
#thus the ARCH model itself is far from enough though the ARCH effect exists


##GARCH LOOP
aic_all2 = c()

for (i in 1:4){
  for (j in 0:3){
```

```
        fit_garch_loop=garchFit(substitute(~ arma(4,5)+garch(p,q),list(p=i, q=j)),dat
a=tb_train_diff_transformed,trace=F)

        aic2 = fit_garch_loop@fit$ics[1]
        aic_tmp2 = c(i,j,aic2)
        aic_all2 = rbind(aic_all2,aic_tmp2)

  }
}

z2=which.min(aic_all2[,3])
combos2=cbind(1:16,aic_all2)
combos2[z2,]
#                 i         j         AIC
#2.000000  1.000000  1.000000 -2.077399

#TEST
set.seed(777)
fit777=garchFit(~arma(4,5)+garch(1,1),data=tb_train_diff_transformed,trace=F)
summary(fit777)
# AIC       BIC       SIC       HQIC
#-2.077399 -1.970265 -2.078632 -2.035413
mae.cob<-mean(abs(fit777@residuals))
mse.cob<-mean((fit777@residuals)^2)
mape.cob<-mean(abs((fit777@residuals)/tb_train_diff_transformed))*100
cbind(mae.cob,mse.cob,mape.cob)
#  mae.cob    mse.cob mape.cob
#[1,] 0.06922825 0.00970512      Inf

#test for Arch effect
fit777.residuals <- fit777@residuals
library(fpp2)
ggAcf(fit777.residuals)
pacf(fit777.residuals^2)
log(538)
#6.287859
#pred.garch1=predict(fit777,21)
#ARCH effect exists. Garch model can be used.

#forecast
#install.packages("rugarch")
library(rugarch)
library(fGarch)
spec <- ugarchspec(variance.model = list(model = "sGARCH",
                                         garchOrder = c(1, 2),
                                         submodel = NULL,
                                         external.regressors = NULL,
                                         variance.targeting = FALSE),

                   mean.model     = list(armaOrder = c(4, 5),
                                         external.regressors = NULL,
                                         distribution.model = "norm",
                                         start.pars = list(),
```

```
                                        fixed.pars = list()))

garch <- ugarchfit(spec = spec, data = tb_train_diff_transformed, solver.control = li
st(trace=0))
pred.garch<-ugarchforecast(garch,n.ahead = 22)

pred.garch1<- ts(fitted(pred.garch), frequency=12, start=c(2018, 1))
pred.garch1
###Backtransforming GARCH
final_value<- tb_train_transformed [length(tb_train_transformed)]
pred.garch1[1] <- final_value+pred.garch1[1]
for (i in 2:22) {
  pred.garch1[i] <- pred.garch1[i]+ pred.garch1[i-1]
  }
plot.garch1<-plot(pred.garch1)
plot.garch1


GARCH.mspe=mean((pred.garch1- tb_test_diff_transformed)^2)
accuracy(pred.garch1,tb_test_diff_transformed)
GARCH.mspe
#0.00523707
(fitted(pred.garch))
summary(fit777)
#AIC       BIC       SIC       HQIC
#-2.077399 -1.970265 -2.078632 -2.035413
garch.aic <- -2.077399
garch.rmse <- sqrt(mean((fitted(pred.garch)- tb_test_diff_transformed)^2))
garch.mape <-  100*(mean(abs((fitted(pred.garch)- tb_test_diff_transformed)/tb_test_d
iff_transformed)))
garch.mae <- mean(abs(fitted(pred.garch)- tb_test_diff_transformed))
#garch.mase <- mean(abs(fitted(pred.garch)-tb_test_transformed)/sum())
cbind(garch.mae,garch.rmse,garch.mape)
#    garch.mae garch.rmse garch.mape
#[1,]  0.052223 0.07237151  -33.38917
checkresiduals(fit777@residuals)
Box.test(fit777@fitted)
# Box-Pierce test
#
# data:  fit777@fitted
# X-squared = 16.364, df = 1, p-value = 5.227e-05

###Forecast for 1QFY2020###
library(rugarch)
library(fGarch)
spec <- ugarchspec(variance.model = list(model = "sGARCH",
                                         garchOrder = c(1, 2),
                                         submodel = NULL,
                                         external.regressors = NULL,
                                         variance.targeting = FALSE),

                   mean.model     = list(armaOrder = c(4, 5),
                                         external.regressors = NULL,
                                         distribution.model = "norm",
                                         start.pars = list(),
```

```
                                          fixed.pars = list()))

garch <- ugarchfit(spec = spec, data = tb_train_diff_transformed, solver.control = li
st(trace=0))
pred.garch_1q<-ugarchforecast(garch,n.ahead = 27)

pred.garch1_1q<- ts(fitted(pred.garch_1q), frequency=12, start=c(2018, 1))
pred.garch1_1q
###Backtransforming GARCH
final_value<- tb_train_transformed [length(tb_train_transformed)]
pred.garch1_1q[1] <- final_value+pred.garch1_1q[1]
for (i in 2:27) {
  pred.garch1_1q[i] <- pred.garch1_1q[i]+ pred.garch1_1q[i-1]
}
(pred.garch1_1q)
(r.hat6_1q<-predict(fit6,27)$pred)

plot(pred.garch1_1q,col='cyan4', main = "Forecast of T-Bill rates through 1Q2020", yl
ab = 'in percentage', xlab='')
lines(r.hat6_1q,col='chocolate1')
legend('topleft',c('pred.ARIMA 1Q', 'pred.garch 1Q'),lty=rep(1,5),
       col=c('chocolate1','cyan4'),cex = 0.5)
```

SETAR MODEL

```
#install.packages('tsDyn')
library(tsDyn)
fit8=linear(tb_train_transformed, m=1)#linear model
summary(fit8)
fit9= aar(tb_train_transformed, m=1) #non-linear additive AR model
summary(fit9)
Box.test(fit8$residuals,lag=10,type='Ljung')
Box.test(fit9$residuals,lag=10,type='Ljung')

AIC(fit8) #-2285.158
AIC(fit9) #-2269.158; linear model is better


fit.setar1<-setar(tb_train_transformed, m=2, thDelay=1)
summary(fit.setar1)
AIC(fit.setar1) #-2374.759
acf(fit.setar1$residuals) #plot: project_setar1
Box.test(fit.setar1$residuals,lag=10,type='Ljung') # 3.835e-06
pred1=predict(fit.setar1,n.ahead=22)

mean((pred1-as.numeric(tb_test_transformed))^2) #MSE: 0.1812498
accuracy(pred1,tb_test_transformed)
# ME       RMSE       MAE       MPE      MAPE       ACF1 Theil's U
# Test set 0.3877714 0.4257344 0.3877714 43.21572 43.21572 0.7980753  5.051177
checkresiduals(fit.setar1)

fit.setar2 <- setar(tb_train_transformed, m=3, thDelay=1)
summary(fit.setar2)
AIC(fit.setar2) # -2394.382
acf(fit.setar2$residuals) #plot: project_setar2
Box.test(fit.setar2$residuals,lag=10,type='Ljung') #0.0004806
```

```
pred2=predict(fit.setar2,n.ahead=22)
mean((pred2-as.numeric(tb_test_transformed))^2) #MSE:  0.2749303


fit.setar4 <- setar(tb_train_transformed, m=4, thDelay=1)
summary(fit.setar4)
AIC(fit.setar4) #-2402.722
acf(fit.setar4$residuals) #plot: project_setar4
Box.test(fit.setar4$residuals,lag=10,type='Ljung') #0.01163
pred4=predict(fit.setar4,n.ahead=22)
mean((pred4-as.numeric(tb_test_transformed))^2) #MSE: 0.2237386

fit.setar5 <- setar(tb_train_transformed, m=5, thDelay=1)
summary(fit.setar5)
AIC(fit.setar5) #-2404.35, only improved by a slight amount. Selecting fit.setar4
acf(fit.setar4$residuals) #plot: project_setar4
Box.test(fit.setar4$residuals,lag=10,type='Ljung') # 0.00919
pred4=predict(fit.setar5,n.ahead=22)
mean((pred4-as.numeric(tb_test_transformed))^2) #MSE: 0.2558239

setar.aic <- -2402.722
setar.mae<-mean(abs(fit.setar4$residuals))
setar.rmse<-sqrt(mean((fit.setar4$residuals)^2))
setar.mape<-mean(abs((fit.setar4$residuals)/tb_train_transformed[1:516]))*100
c(setar.mae,setar.rmse,setar.mape)
#[1] 0.06464902 0.08821089 8.38877628
```

LSTAR MODEL

```
fit.lstar <- lstar(tb_train_transformed, m=4, thDelay=1)
summary(fit.lstar)
AIC(fit.lstar) #-2400.388
lstar.aic <- -2400.388
checkresiduals(fit.lstar$residuals)
pred.lstar <- predict(fit.lstar, n.ahead=22)
Box.test(fit.lstar$residuals,lag = 9, type = "Ljung")
#Box-Ljung test

#data:  fit.lstar$residuals
#X-squared = 21.408, df = 9, p-value = 0.01096
accuracy(pred.lstar, tb_test_transformed)
lstar.rmse <- 0.4782705
lstar.mae <- 0.4416052
lstar.mape <- 49.96487
plot(1:22,tb_test_transformed,type='l',col='black',lwd=2,xlim=c(1,22),xlab='Month',yl
ab = 't-bill rate')
lines( 1:22, pred.lstar, type="o",pch=24, col="red",lwd=2)
```

Neural Network Model

```
set.seed(11)
fit.nn<-nnetar(tb_train_transformed)
summary(fit.nn)
pred.nn=forecast(fit.nn,22)
plot(1:22,tb_test_transformed,type='l',col='black',lwd=2,xlim=c(1,22),xlab='Month',yl
ab = 't-bill rate')
```

```r
lines(1:22,pred.nn$mean, type="o",pch=24, col="red",lwd=2)
accuracy(pred.nn,tb_test_transformed)
nn.rmse <- 0.72675805
nn.mae <- 0.53558385
nn.mape <- 0.53558385
nn.mase <- -73.245918
#  ME        RMSE         MAE         MPE       MAPE       MASE         ACF1 Theil's U
# Training set  0.0001973782 0.07235805 0.05513584  -3.910303  8.847313 0.1310221 -0.
03272649        NA
# Test set     -0.5308377243 0.72675805 0.53558385 -73.245918 73.927265 1.2727354  0.
86170642  11.39644

checkresiduals(fit.nn)
fit.nn%>%forecast(22)%>%autoplot()
Box.test(fit.nn$residuals,lag = 9, type='Ljung')
# Box-Ljung test
#
# data:  fit.nn$residuals
# X-squared = 1.55, df = 9, p-value = 0.9968
```

regression

```r
head(tb.rates)
length(tb_test_transformed)
unemployment <- ts(tb.rates$unemployment, frequency=12, start=c(1975, 1))
unemployment_train1 <- window(unemployment, end = c(2017,12))
unemployment_test1<-window(unemployment, start=c(2018, 1))
return <- ts(tb.rates$stock_return_nominal , frequency=12, start=c(1975, 1))
return_train1 <- window(return, end = c(2017,12))
return_test1<-window(return, start=c(2018, 1))
#install.packages("MASS")
head(cu_train1)
library(MASS)
reg1 <- tslm(tb_train_transformed~unemployment_train1+return_train1)
AIC(reg1)
summary(reg1)
reg2 <- tslm(diff(tb_train_transformed)~diff(unemployment_train1)+diff(return_train
1))
summary(reg2)
AIC(reg2)
df <-data.frame(unemployment_test1,return_test1)
names <-c("unemployment_train1","return_train1")
colnames(df) <- names
fcast <- predict.lm(reg1, newdata =df,h=22)
plot(fcast)
checkresiduals(reg2)
checkresiduals(reg1)
#residuals are not white noise
accuracy(fcast,tb_test_transformed)
```

prediction graph

```r
# ETS FIT4
#ARIMA r.hat6
# GARCH pred.garch$meanForecast
#setar pred4
 #lstar pred.lstar
```

```r
 #network pred.nn

par(mfrow=c(1,1))
plot(tb_test_transformed,col='black',xlab = '', ylab = 'in percentage',ylim=c(0,2),ma
in='Test Set Forecast Comparison')
lines(fit4$mean,type="l",pch=2,col="purple")
lines( r.hat6, type="l",pch=2, col="chocolate1")
lines(pred.garch1, type="l",pch=2, col="cyan4")
lines( pred4, type="l",pch=24, col="green",lwd = 0.5)
lines( pred.lstar, type="l",pch=24, col="blue",lwd=2)
lines( pred.nn$mean, type="l",pch=24, col="red",lwd=2)
lines(fcast,type="l",pch=24,col="yellow",lwd=2)
legend('topleft',c('True', 'pred.ets','pred.ARIMA', 'pred.garch', 'pred.setar',
                  'pred.lstar','pred.nn','pred.regression'),lty=rep(1,5),
       col=c('black','purple','chocolate1', 'cyan4','green','blue','red','yellow'),ce
x = 0.5)

cbind(ets.aic,arima.aic,garch.aic,setar.aic,lstar.aic,reg.aic)
#ets.aic arima.aic garch.aic setar.aic lstar.aic  reg.aic
#914.2691 -919.7686 -2.077399  -2404.35 -2400.388 1059.271

cbind(ets.mae,arima.mae,garch.mae,setar.mae,lstar.mae,reg.mae)
#ets.mae arima.mae garch.mae  setar.mae lstar.mae  reg.mae
#0.1769383 0.1763157 0.6039189 0.06464902 0.4416052 2.464469

cbind(ets.mape,arima.mape,garch.mape, setar.mape,lstar.mape,reg.mape)

cbind(ets.rmse,arima.rmse,setar.rmse,lstar.rmse,reg.rmse)
#ets.rmse arima.rmse garch.rmse setar.rmse lstar.rmse reg.rmse
#0.1948937  0.1901198 0.07237151  0.08821089  0.4782705 2.473038
```